Detecting Fraudulent Credit Card Transactions Using Logistic Regression Classification.

B. Mendoza

1 Introduction

The purpose of this project is to detect fraudulent credit card transactions. Credit card fraud is a prevalent issue around the world. According to the Federal Trade Commission's (FTC) online database of consumer complaints, the United States alone saw 1.3 million reports of fraud in 2016, of which 33% were credit card related. [FTC, 2017] Despite this, only a small percentage of total credit card transactions are fraudulent in any given year. Thus, any method of detecting credit card fraud must be robust to severely imbalanced data.

We primarily study logistic regression classification as a tool for fraud detection. The use of logistic regression is motivated by our imbalanced dataset and our desire to model the probability of being fraudulent given the transaction data. The GitHub repository for the project can be found here.

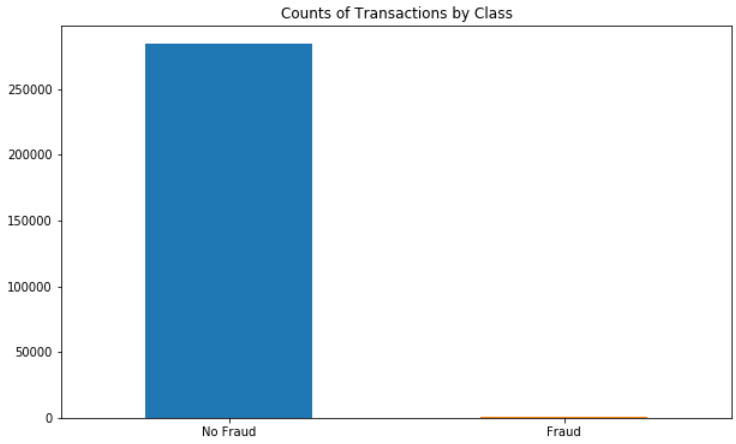

Figure 1: Imbalance between fraudulent and non-fraudulent transactions in the ULB dataset

2 Methodology

2.1 Acquiring Data and Dataset Description

Credit card transaction data was acquired from a public dataset hosted by the Machine Learning Department at the Université Libre de Bruxelles (ULB). The transactions are from European cardholders and span a time period of two days in September 2013. As visualized in Figure 1, the dataset is highly imbalanced, with only 492 frauds out of 284,807 total transactions. This is a fraud rate of 0.17%. The data can be found here, and descriptive statistics are listed in the Appendix.

The ULB dataset consists of 30 predictor variables and one binary response variable named "Class" (taking the value 0 for non-fraudulent data and 1 for fraudulent data). Of

the 30 predictor variables, 28 are the result of a Principal Component Analysis performed by ULB prior to publicizing the dataset. These variables are labeled "V1-V28." ULB states that the purpose of this transformation is to anonymize any identifying information in the transactions such as location, account number, etc. We will also see that the analysis of these principal components yields some underlying structure to the transaction data.

The remaining two predictors are "Time" and "Amount," where "Time" refers to the number of seconds between each transaction and the first transaction in the dataset. "Amount" densities are shown in Figure 2, which demonstrate that the majority of transactions are small regardless of "Class." It is possible that the distribution of fraudulent "Amount" would converge to the distribution of non-fraudulent "Amount" given enough fraudulent transactions, though there is no way of knowing for certain.

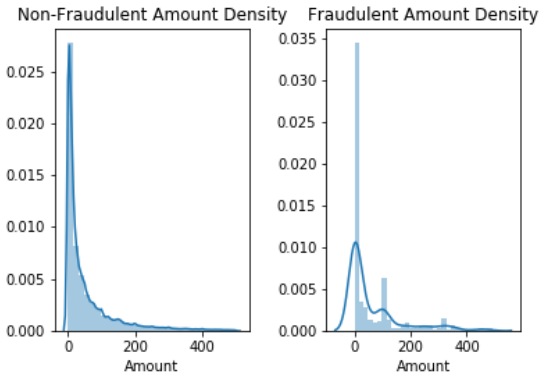

Figure 2: Density plots of "Amount" by "Class"

Since principal components are the eigenvectors of the covariance matrix of parameters, they are orthogonal by nature. Thus, the majority of our variables are uncorrelated. Figure 3 shows the sparsity of our correlation matrix other than in the "Time," "Amount," and "Class" variables. This sparsity reduces the variance of our regression coefficients.

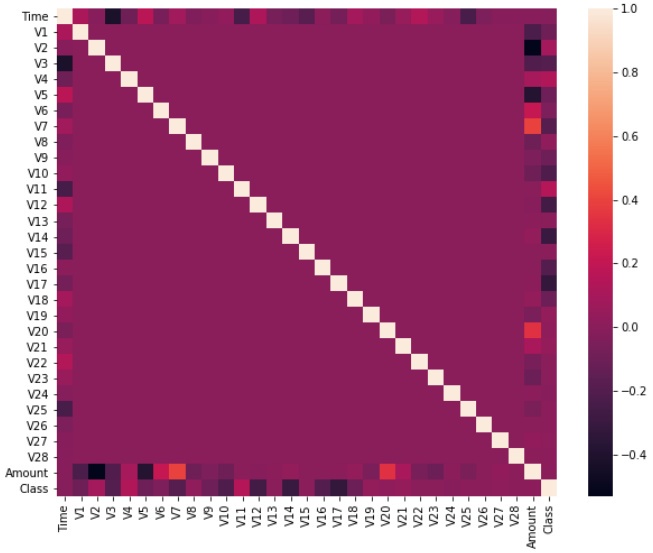

Figure 3: Correlation Matrix

Our variables "V1" and "V2" are the first two principal components of our dataset. Since they account for the majority of the variability in our data, we can plot a projection onto the "V1×V2" plane for a 2-D visualization of the data. Figure 4 indicates a planar structure in the data. This is likely due to other numerical columns being similar to "Amount" in distribution (clumping near zero, sparse on larger magnitudes). PCA would not be able to undo this clumping, which would lead to much of the data being concentrated in a planar structure. This structure becomes linear when projected onto two dimensions.

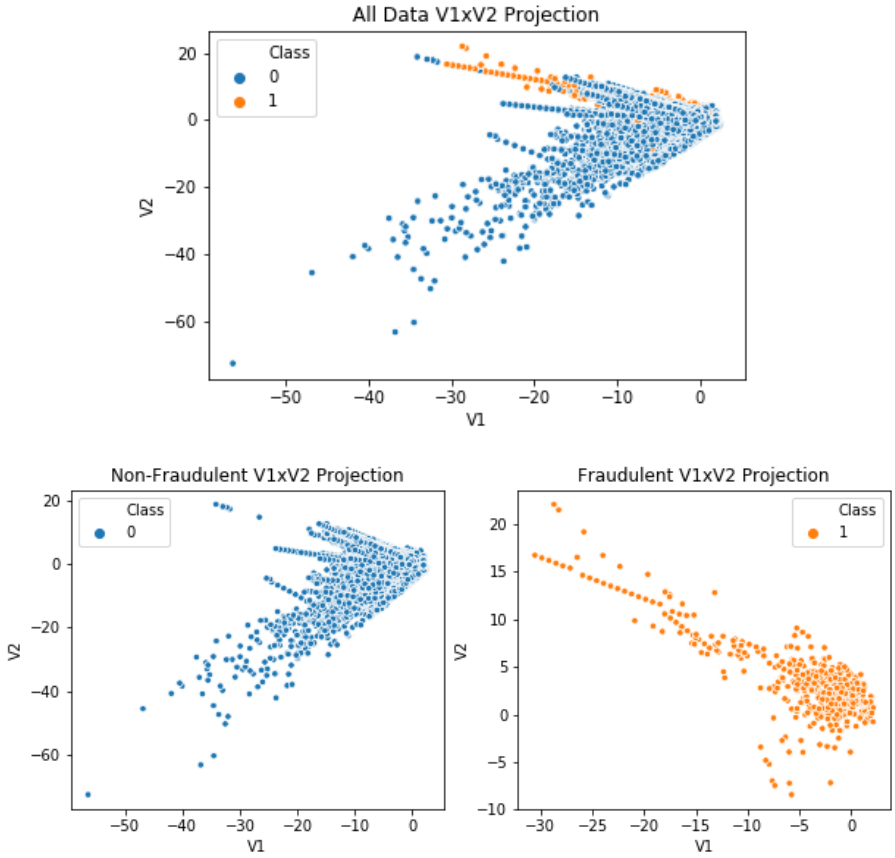

Figure 4: *(Top)* Full data projection onto "V1×V2"
(Left) Non-fraud data projection onto "V1×V2"
(Right) Fraud projection onto "V1×V2"

2.2 Justification for Logistic Regression

If we let random variable X represent our predictor columns and random variable Y represent our response column, then our classification problem becomes seeing a vector $x \in \mathbb{R}^p$ and trying to guess a $y \in \{0, 1\}$ using a classification function $f(X)$. In our specific dataset, $p = 30$ is the number of parameters. It is well-known in statistical decision theory that for 0-1 loss, the classifier with the best possible misclassification rate is the Bayes classifier: [Yu, 2013]

$$\hat{f}(x) = \arg\max_{j=0,1} P(Y = j \mid X = x).$$

This classifier predicts the most likely class given the data $x \in \mathbb{R}^p$. However, one can not look at $P(Y = j \mid X = x)$ directly for a large number of possible x (in our case x is essentially continuous). This motivates us to pool information from similar points in a regression-like manner.

When examining the conditional probability of being fraudulent, one might opt for a linear model $P(Y = 1 \mid X = x) = \beta^T x$. However, a linear model could predict a value outside of $[0, 1]$, which would nonsensical for probabilities. Furthermore, the interpretation of regression coefficients is not intuitive. Despite this, we still wish to preserve the convenience of forming linear functions $\beta^T x$ of our data, and logistic regression provides a solution by instead modeling

$$\log\left(\frac{P(Y = 1 \mid X = x)}{P(Y = 0 \mid X = x)}\right) = \beta^T x.$$

Here, we are modeling the log-odds (logit) of being fraudulent as linear. This turns out to be much more natural, as the inverse logit will send values exclusively into $[0, 1]$ in compliance with probability.

$$P(Y = 1 \mid X = x) = \frac{\exp(\beta^T x)}{1 + \exp(\beta^T x)}.$$

Under this paradigm, one can naturally interpret regression coefficients and also set decision boundaries for classification.

2.3 Training the Classification Model

Simple random sampling was utilized to perform a training-testing split of 67/33 (190,821/93,986). A full logistic regression model was initially trained with no regularization and evaluated on all 30 predictors. Variable selection was then performed with a simplistic selection criterion (keep the 18 coefficients with $p < 0.05$) and the new sparse model was retrained and evaluated. Evaluation criterion include misclassification rate, top-1000 candidates score, and AUC score. A baseline misclassification error (guess everything as non-fraud) was also established for comparison. Regression coefficients and p-values are listed in the Appendix.

3 Results

In the testing set, a simple baseline misclassification rate was established at 0.001574 by guessing every point as non-fraudulent. Both the full and sparse models were able to outperform this naive classifier, but this indicates that misclassification rate is not a robust metric for such an imbalanced dataset.

	Misclassification Rate
All 0	0.001574
Full Model	0.001096
Sparse Model	0.000958

Table 1: Misclassification rate for each model

The top-1000 most-likely-to-be-fraud transactions were then selected from the testing set based on the predicted probabilities of our models. The full model was able to place 97 actually fraudulent transactions into its top-1000 and the sparse model was able to place 107. For comparison, the testing set had only 148 fraudulent transactions out of 93,986. A random sample of 1000 from 93,986 would only be expected to contain 1.5 frauds (hypergeometric distribution).

	Top-1000 Score
Full Model	9.7%
Sparse Model	10.7%

Table 2: Top 1000 most-likely-to-be-fraud transactions true positive rate for each model

ROC (sensitivity versus specificity) curves were plotted for both models as seen in Figure 5. AUC (area-under-curve) scores were also calculated for the respective curves.

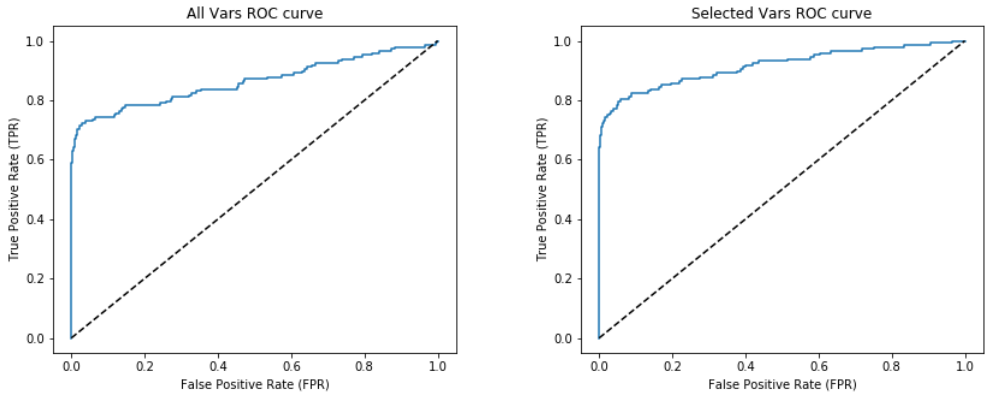

Figure 5: *(Left)* Full model ROC curve
(Right) Sparse model ROC curve

	AUC Score
Full Model	0.863420
Sparse Model	0.916065

Table 3: AUC score for each model

4 Evaluation

Even though our classifiers do not significantly outperform "All 0" in terms of 0-1 loss, we see that they can help to sort our data according to their likelihood of being fraud. This can be a much more reasonable goal than misclassification error in heavily imbalanced samples. One can think of this as an approach to selecting an "enriched" set. As we change the size of our enriched set, we change both:

- The fraction of true positives in the data that one manages to capture (sensitivity).
- The fraction of true negatives one manages to avoid (specificity).

When the set gets larger, sensitivity will necessarily increase, but specificity will decrease. Our sparse classifier makes this tradeoff more efficiently than our full model, as evidenced by the AUC scores. This is an indication that the full classifier may be overfitting to less important variables that are selected out by our $p < 0.05$ criterion. We can see this tradeoff in action by examining histograms of the predicted logits for the two models.

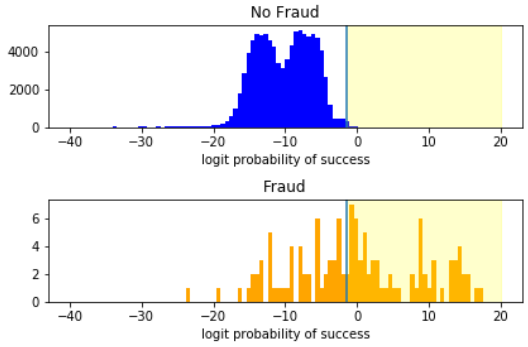

Figure 6: Histograms of predicted logits for full model by class

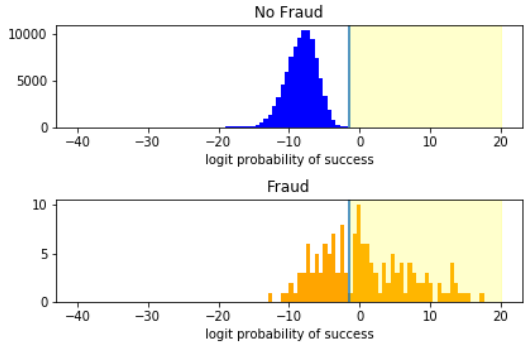

Figure 7: Histograms of predicted logits for sparse model by class

Note that the logits have different distributions between classes. We are able to use a logit (or probability) cutoff to classify due to the wider distribution of fraudulent data. In Figure 6/7, a logit cutoff of $\beta^T x = -1.5$ is demonstrated, corresponding to a probability cutoff of $P(Y = 1 \mid X = x) = 0.182$. We can remark that:

- The shaded regions in the "No Fraud" histograms correspond to False Positive observations (observations that were labeled fraud but were actually not) and the proportion of observations in this region is the False Positive Rate (FPR). We have Specificity $= 1 -$ FPR.
- The shaded regions in the "Fraud" histograms correspond to True Positive observations (observations that were labeled fraud correctly), and the proportion of observations in this region is the True Positive Rate (TPR). We have Sensitivity = TPR.

5 Conclusions and Future Work

Logistic regression can be a powerful tool for classification. It is an intuitive way to model the predicted probability of being in a class while being robust to large imbalances in the data. Even if one does not wish to use logistic regression for predictions, one can still form "enriched" datasets by sorted rows in terms of $P(Y = 1 \mid X = x)$.

Further work can include examining more complex variable selection criterion and analyzing other models entirely. Our work was constrained to predict with principal components, but one unfortunate consequence is a decrease in interpretability. It could hold that our logit values are not well-predicted by the principal components but would be well-predicted by some other combination of the original variables. [Miller, 2002]

References

[FTC, 2017] FTC (2017). Consumer sentinel network data book for january - december 2016.

[Miller, 2002] Miller, A. J. (2002). Subset selection in regression.

[Yu, 2013] Yu, A. J. (2013). Introduction to bayesian decision theory.

6 Appendix

6.1 Summary Statistics

	count	mean	std	min	25%	50%	75%	max
Time	284807.000	94813.860	47488.146	0.000	54201.500	84692.000	139320.500	172792.000
V1	284807.000	0.000	1.959	-56.408	-0.920	0.018	1.316	2.455
V2	284807.000	0.000	1.651	-72.716	-0.599	0.065	0.804	22.058
V3	284807.000	-0.000	1.516	-48.326	-0.890	0.180	1.027	9.383
V4	284807.000	0.000	1.416	-5.683	-0.849	-0.020	0.743	16.875
V5	284807.000	0.000	1.380	-113.743	-0.692	-0.054	0.612	34.802
V6	284807.000	0.000	1.332	-26.161	-0.768	-0.274	0.399	73.302
V7	284807.000	-0.000	1.237	-43.557	-0.554	0.040	0.570	120.589
V8	284807.000	0.000	1.194	-73.217	-0.209	0.022	0.327	20.007
V9	284807.000	-0.000	1.099	-13.434	-0.643	-0.051	0.597	15.595
V10	284807.000	0.000	1.089	-24.588	-0.535	-0.093	0.454	23.745
V11	284807.000	0.000	1.021	-4.797	-0.762	-0.033	0.740	12.019
V12	284807.000	-0.000	0.999	-18.684	-0.406	0.140	0.618	7.848
V13	284807.000	0.000	0.995	-5.792	-0.649	-0.014	0.663	7.127
V14	284807.000	0.000	0.959	-19.214	-0.426	0.051	0.493	10.527
V15	284807.000	0.000	0.915	-4.499	-0.583	0.048	0.649	8.878
V16	284807.000	0.000	0.876	-14.130	-0.468	0.066	0.523	17.315
V17	284807.000	-0.000	0.849	-25.163	-0.484	-0.066	0.400	9.254
V18	284807.000	0.000	0.838	-9.499	-0.499	-0.004	0.501	5.041
V19	284807.000	0.000	0.814	-7.214	-0.456	0.004	0.459	5.592
V20	284807.000	0.000	0.771	-54.498	-0.212	-0.062	0.133	39.421
V21	284807.000	0.000	0.735	-34.830	-0.228	-0.029	0.186	27.203
V22	284807.000	-0.000	0.726	-10.933	-0.542	0.007	0.529	10.503
V23	284807.000	0.000	0.624	-44.808	-0.162	-0.011	0.148	22.528
V24	284807.000	0.000	0.606	-2.837	-0.355	0.041	0.440	4.585
V25	284807.000	0.000	0.521	-10.295	-0.317	0.017	0.351	7.520
V26	284807.000	0.000	0.482	-2.605	-0.327	-0.052	0.241	3.517
V27	284807.000	-0.000	0.404	-22.566	-0.071	0.001	0.091	31.612
V28	284807.000	-0.000	0.330	-15.430	-0.053	0.011	0.078	33.848
Amount	284807.000	88.350	250.120	0.000	5.600	22.000	77.165	25691.160
Class	284807.000	0.002	0.042	0.000	0.000	0.000	0.000	1.000

6.2 Coefficients and p-values for Full Model

Variable	Coefficient	p-value	Selected
Intercept	-0.87	0	·
Time	-9.01e-5	0	·
V1	1.33e-1	7.11e-6	·
V2	-2.88e-1	2.31e-11	·
V3	-7.45e-1	0	·
V4	1.09e-1	4.81e-4	·
V5	-5.94e-2	1.95e-1	
V6	-7.28e-2	4.53e-2	·
V7	7.90e-3	8.895e-1	
V8	-9.08e-2	7.90e-3	·
V9	-3.75e-1	0	·
V10	-2.26e-1	8.45e-7	·
V11	-1.29e-1	1.28e-3	·
V12	-2.34e-2	5.34e-1	
V13	-2.50e-1	2.02e-10	·
V14	-6.66e-1	0	·
V15	-2.04e-1	7.08e-5	·
V16	-2.61e-1	1.42e-7	·
V17	-4.45e-1	0	·
V18	-2.22e-2	6.71e-1	
V19	4.95e-2	3.78e-1	
V20	5.80e-2	5.24e-1	
V21	1.43e-1	1.04e-2	·
V22	1.77e-1	1.62e-2	·
V23	2.41e-2	7.63e-1	
V24	-2.71e-2	6.75e-1	
V25	-1.47e-1	1.55e-1	
V26	2.60e-2	7.74e-1	
V27	-5.12e-2	6.73e-1	
V28	3.09e-2	6.53e-1	
Amount	-7.77e-3	0	·

Table 4: Coefficient and p-values for full model

Coefficients are interpreted in the standard way. For instance, our full model predicts a 0.133 increase in the log-odds of being fraudulent for every unit increase in "V1."

6.3 Coefficients and p-values for Sparse Model

Variable	Coefficient	p-value	Selected
Intercept	-3.30	0	·
Time	-4.93e-5	0	·
V1	3.99e-1	0	·
V2	-4.76e-1	0	·
V3	-6.19e-1	0	·
V4	1.45e-1	3.54e-4	·
V6	-1.07e-1	1.41e-2	·
V8	-3.50e-1	0	·
V9	-3.38e-1	8.24e-9	·
V10	-1.84e-1	5.10e-3	·
V11	-3.65e-1	6.59e-12	·
V13	-3.20e-1	7.33e-9	·
V14	-6.88e-1	0	·
V15	-5.27e-1	0	·
V16	-2.05e-1	6.11e-4	·
V17	-1.97e-1	9.11e-6	·
V21	4.40e-1	0	·
V22	6.88e-1	7.62e-14	·
Amount	-3.00e-3	5.17e-11	·

Table 5: Coefficient and p-values for sparse model

Coefficients are interpreted in the standard way. For instance, our sparse model predicts a 0.399 increase in the log-odds of being fraudulent for every unit increase in "V1."